Your search matched **14** of **983096** documents.

A maximum of **14** results are displayed, **15** to a page, sorted by **Relevance** in **descending** order.
You may refine your search by editing the current search expression or entering a new one the text
Then click **Search Again**.

((type)and (hierarchy)) and(checking)

Search Again

Results:
Journal or Magazine = **JNL** Conference = **CNF** Standard = **STD**

1 **A type system for an object-oriented database system**
Wu, X.;
Computer Software and Applications Conference, 1991. COMPSAC '91., Proceed
the Fifteenth Annual International , 11-13 Sept. 1991
Page(s): 333 -338

[Abstract] [PDF Full-Text (444 KB)] **IEEE CNF**

2 **The requirement model in a knowledge-based rapid prototyping syste**
Chen, P.-M.; Chou, C.-R.;
Computer Software and Applications Conference, 1988. COMPSAC 88. Proceedir
Twelfth International , 5-7 Oct. 1988
Page(s): 418 -426

[Abstract] [PDF Full-Text (996 KB)] **IEEE CNF**

3 **Extending high-level Petri nets for object-oriented design**
Camurri, A.; Franchi, P.; Vitale, M.;
Systems, Man and Cybernetics, 1992., IEEE International Conference on , 18-2:
1992
Page(s): 623 -626 vol.1

[Abstract] [PDF Full-Text (228 KB)] **IEEE CNF**

4 **Object state testing for object-oriented programs**
Gao, J.Z.; Kung, D.; Hsia, P.; Toyoshima, Y.; Chen, C.;
Computer Software and Applications Conference, 1995. COMPSAC 95. Proceedir
Nineteenth Annual International , 9-11 Aug. 1995

Page(s): 232 -238

[Abstract] [PDF Full-Text (556 KB)] **IEEE CNF**

5 **Preserving consistency in nested transactions**
Schoning, H.;
System Sciences, 1990., Proceedings of the Twenty-Third Annual Hawaii Interna
Conference on , Volume: ii , 2-5 Jan. 1990
Page(s): 472 -480 vol.2

[Abstract] [PDF Full-Text (604 KB)] **IEEE CNF**

6 **On the interaction between ISA and cardinality constraints**
Calvanese, D.; Lenzerini, M.;
Data Engineering, 1994. Proceedings. 10th International Conference , 14-18 Fel
Page(s): 204 -213

[Abstract] [PDF Full-Text (716 KB)] **IEEE CNF**

7 **Formal representation of reuseable software modules**
Tjandra, I.A.; Butler, G.;
Reverse Engineering, 1995., Proceedings of 2nd Working Conference on , 14-16
1995
Page(s): 198 -205

[Abstract] [PDF Full-Text (604 KB)] **IEEE CNF**

8 **Interactive verification of switched-current sigma-delta modulators**
de la Rosa, J.M.; Kaiser, A.; Perez-Verdu, B.;
Electronics, Circuits and Systems, 1998 IEEE International Conference on , Volu
7-10 Sept. 1998
Page(s): 157 -160 vol.2

[Abstract] [PDF Full-Text (292 KB)] **IEEE CNF**

9 **A framework for dynamic checks of C arrays via a C++ class**
Macphee, K.E.; Dix, T.I.;
Software Engineering Conference, 1998. Proceedings. 1998 Australian , 9-13 Nc

Page(s): 57 -65

[Abstract] [PDF Full-Text (184 KB)] **IEEE CNF**

10 **Weak subtyping-yet another notion of inheritance**
Lowe, W.; Neumann, R.; Trapp, M.; Zimmermann, W.;
Technology of Object-Oriented Languages, 1998. TOOLS 26. Proceedings , 3-7 A
1998
Page(s): 333 -345

[Abstract] [PDF Full-Text (80 KB)] **IEEE CNF**

11 **Higher-dimensional Hadamard matrices**
Shlichta, P.;
Information Theory, IEEE Transactions on , Volume: 25 Issue: 5 , Sep 1979
Page(s): 566 -572

[Abstract] [PDF Full-Text (1072 KB)] **IEEE JNL**

12 **Concurrent detection of software and hardware data-access faults**
Wilken, K.D.; Kong, T.;
Computers, IEEE Transactions on , Volume: 46 Issue: 4 , April 1997
Page(s): 412 -424

[Abstract] [PDF Full-Text (336 KB)] **IEEE JNL**

13 **Existence dependency: The key to semantic integrity between structi and behavioral aspects of object types**
Snoeck, M.; Dedene, G.;
Software Engineering, IEEE Transactions on , Volume: 24 Issue: 4 , April 1998
Page(s): 233 -251

[Abstract] [PDF Full-Text (604 KB)] **IEEE JNL**

14 **Generalized Hamming weights of nonlinear codes and the relation to Z_4-linear representation**
Reuven, I.; Be'ery, Y.;
Information Theory, IEEE Transactions on , Volume: 45 Issue: 2 , March 1999
Page(s): 713 -720

[Abstract] [PDF Full-Text (320 KB)] **IEEE JNL**

Search Results

Search Results for: **[inherit <near> tree <near> checking <AND>((type <near> class <near> hierarchy <near> object <near> cache <near> tree))]**
Found **15** of **122,228 searched.**

Search within Results

Results 1 - 15 of 15 short listing

1 Time and space efficient method-lookup for object-oriented programs 13%
S. Muthukrishnan , Martin Müller
Proceedings of the seventh annual ACM-SIAM symposium on Discrete algorithms
January 1996

2 Oblog-2: A hybrid knowledge representation system for defeasible reasoning 8%
T. F. Gordon
Proceedings of the first international conference on Artificial intelligence and law
December 1987
Oblog-2 is a hybrid knowledge representation system comparable to Krypton and KL-TWO. It combines a terminological reasoner with a Prolog-like inference mechanism. The terminological component supports the description of type and attribute taxonomies. Entities are instances of a set of types. Procedures for determining the values of attributes are Horn clause rules indexed by type. The known types of an entity determine its set of applicable rules, which changes as our knowledge about the t ...

3 Efficient handling of multiple inheritance hierarchies 5%
Yves Caseau
ACM SIGPLAN Notices , Proceedings of the eighth annual conference on Object-oriented programming systems, languages, and applications October 1993
Volume 28 Issue 10

4 BlueBoX: A policy-driven, host-based intrusion detection system 4%
Suresh N. Chari , Pau-Chen Cheng
ACM Transactions on Information and System Security (TISSEC) May 2003

Volume 6 Issue 2
Detecting attacks against systems has, in practice, largely been delegated to sensors, such as network intrustion detection systems. However, due to the inherent limitations of these systems and the increasing use of encryption in communication, intrusion detection and prevention have once again moved back to the host systems themselves. In this paper, we describe our experiences with building BlueBox, a host-based intrusion detection system. Our approach, based on the technique of system call i ...

5 Fast ray tracing by ray classification 4%
James Arvo , David Kirk
ACM SIGGRAPH Computer Graphics , Proceedings of the 14th annual conference on Computer graphics and interactive techniques August 1987
Volume 21 Issue 4

6 Selector table indexing & sparse arrays 1%
Karel Driesen
ACM SIGPLAN Notices , Proceedings of the eighth annual conference on Object-oriented programming systems, languages, and applications October 1993
Volume 28 Issue 10

7 Anatomy of a native XML base management system 1%
T. Fiebig , S. Helmer , C.-C. Kanne , G. Moerkotte , J. Neumann , R. Schiele , T. Westmann
The VLDB Journal — The International Journal on Very Large Data Bases December 2002
Volume 11 Issue 4
Several alternatives to manage large XML document collections exist, ranging from file systems over relational or other database systems to specifically tailored XML base management systems. In this paper we give a tour of Natix, a database management system designed from scratch for storing and processing XML data. Contrary to the common belief that management of XML data is just another application for traditional databases like relational systems, we illustrate how almost every component in a ...

8 Developing a GUIDE using object-oriented programming 1%
Joseph A. Konstan , Lawrence A. Rowe
ACM SIGPLAN Notices , Conference proceedings on Object-oriented programming systems, languages, and applications November 1991
Volume 26 Issue 11

9 Fast algorithm for creating space efficient dispatching tables with application to multi-dispatching 1%
Yoav Zibin , Joseph Yossi Gil
ACM SIGPLAN Notices , Proceedings of the 17th ACM conference on Object-oriented programming, systems, languages, and applications November 2002
Volume 37 Issue 11
The dispatching problem can be solved very efficiently in the single-inheritance~(SI) setting. In this paper we show how to extend one such solution to the multiple-inheritance~(MI) setting. This generalization comes with an increase to the space requirement by a small factor of

κ This factor can be thought of as a metric of the complexity of the topology of the inheritance hierarchy.On a data set of~35 hierarchies totaling some~64 thousand types, our dispatching data structure, based on a ...

10 Sealed calls in Java packages 1%
Ayal Zaks , Vitaly Feldman , Nava Aizikowitz
ACM SIGPLAN Notices , Proceedings of the conference on Object-oriented programming, systems, languages, and applications October 2000
Volume 35 Issue 10
Determining the potential targets of virtual method invocations is essential for inter-procedural optimizations of object-oriented programs. It is generally hard to determine such targets accurately. The problem is especially difficult for dynamic languages such as Java, because additional targets of virtual calls may appear at runtime. Current mechanisms that enable inter-procedural optimizations for dynamic languages, repeatedly validate the optimizations at runtime. This paper addresses this ...

11 Declarative specialization of object-oriented programs 0%
Eugen N. Volanschi , Charles Counsel , Gilles Muller , Crispin Cowan
ACM SIGPLAN Notices , Proceedings of the 1997 ACM SIGPLAN conference on Object-oriented programming systems, languages and applications October 1997
Volume 32 Issue 10
Designing and implementing generic software components is encouraged by languages such as object-oriented ones and commonly advocated in most application areas. Generic software components have many advantages among which the most important is reusability. However, it comes at a price: genericity often incurs a loss of efficiency.This paper presents an approach aimed at reconciling genericity and efficiency. To do so, we introduce declarations to the Java language to enable a programmer to speci ...

12 A structural view of the Cedar programming environment 0%
Daniel C. Swinehart , Polle T. Zellweger , Richard J. Beach , Robert B. Hagmann
ACM Transactions on Programming Languages and Systems (TOPLAS) August 1986
Volume 8 Issue 4
This paper presents an overview of the Cedar programming environment, focusing on its overall structure—that is, the major components of Cedar and the way they are organized. Cedar supports the development of programs written in a single programming language, also called Cedar. Its primary purpose is to increase the productivity of programmers whose activities include experimental programming and the development of prototype software systems for a high-performance personal computer. T ...

13 On understanding types, data abstraction, and polymorphism 0%
Luca Cardelli , Peter Wegner
ACM Computing Surveys (CSUR) December 1985
Volume 17 Issue 4
Our objective is to understand the notion of *type* in programming languages, present a model of typed, polymorphic programming languages that reflects recent research in type theory, and examine the relevance of recent research to the design of practical programming languages. Object-oriented languages provide both a framework and a motivation for exploring the interaction among the concepts of type, data abstraction, and polymorphism, since they extend

the notion of type t ...

14 Interactive type analysis and extended message splitting; optimizing dynamically-typed object-oriented programs 0%

Craig Chambers , David Ungar

ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1990 conference on Programming language design and implementation June 1990

Volume 25 Issue 6

Object-oriented languages have suffered from poor performance caused by frequent and slow dynamically-bound procedure calls. The best way to speed up a procedure call is to compile it out, but dynamic binding of object-oriented procedure calls without static receiver type information precludes inlining. Iterative type analysis and extended message splitting are new compilation techniques that extract much of the necessary type information and make it possib ...

15 Building knowledge base management systems 0%

John Mylopoulos , Vinay Chaudhri , Dimitris Plexousakis , Adel Shrufi , Thodoros Topologlou

The VLDB Journal — The International Journal on Very Large Data Bases

December 1996

Volume 5 Issue 4

Advanced applications in fields such as CAD, software engineering, real-time process control, corporate repositories and digital libraries require the construction, efficient access and management of large, shared knowledge bases. Such knowledge bases cannot be built using existing tools such as expert system shells, because these do not scale up, nor can they be built in terms of existing database technology, because such technology does not support the rich representational structure and infer ...

Results 1 - 15 of 15 short listing